Challenges and Solutions in Optimizing Execution Performance of a Clinical Decision Support-Based Quality Measurement (CDS-QM) Framework

Tyler J Tippetts, Phillip B Warner, MS, Polina V Kukhareva, MPH, David E Shields, Catherine J Staes, MPH, PhD, Kensaku Kawamoto, MD, PhD, MHS
Department of Biomedical Informatics, University of Utah, Salt Lake City, UT

Abstract

Given the close relationship between clinical decision support (CDS) and quality measurement (QM), it has been proposed that a standards-based CDS Web service could be leveraged to enable QM. Benefits of such a CDS-QM framework include semantic consistency and implementation efficiency. However, earlier research has identified execution performance as a critical barrier when CDS-QM is applied to large populations. Here, we describe challenges encountered and solutions devised to optimize CDS-QM execution performance. Through these optimizations, the CDS-QM execution time was optimized approximately three orders of magnitude, such that approximately 370,000 patient records can now be evaluated for 22 quality measure groups in less than 5 hours (approximately 2 milliseconds per measure group per patient). Several key optimization methods were identified, with the most impact achieved through population-based retrieval of relevant data, multi-step data staging, and parallel processing. These optimizations have enabled CDS-QM to be operationally deployed at an enterprise level.

Introduction

Close relationship between clinical decision support (CDS) and quality measurement (QM). CDS and QM are closely related, in that CDS may recommend that X be done if Y is true, while QM may measure if X was done when Y was true in order to assess the quality of the care given. For example, a CDS rule may recommend that aspirin be given if a patient has suffered an acute myocardial infarction, while a QM may measure if aspirin was given when a patient had suffered an acute myocardial infarction. Another commonality between CDS and QM is the goal of improving the quality of care.[1] It has also been reported in the literature that they also share many of the same implementation challenges.[2,3]

Given the close relationship between CDS and QM, there have been increasing efforts to align these two related aspects of quality improvement. Of particular interest from an informatics perspective, the U.S. Office of the National Coordinator for Health IT (ONC) and the Centers for Medicare & Medicaid Services (CMS) are sponsoring a public-private partnership known as the Clinical Quality Framework initiative to develop a common set of harmonized interoperability standards to enable both CDS and QM.[4]

Benefits of a CDS-based framework for QM and prototype implementation. Due to the similar nature of CDS and QM, there are at least two important benefits to using a CDS-based quality measurement (CDS-QM) framework, in which the same approach is leveraged for both CDS and QM purposes. The first important benefit is semantic consistency. Because the same technical framework and core clinical knowledge is re-used across both CDS and QM, clinicians can receive consistent assessments both when receiving CDS at the point of care and obtaining performance feedback through QM. The second important benefit to CDS-based QM is implementation efficiency. By re-using the same underlying infrastructure and content, what may otherwise have involved two separate teams and development efforts can now be completed by a single team and effort. Such de-duplication of effort is particularly important given the significant complexity and attendant resources involved in the development, validation, and maintenance of this type of knowledge resource.[5]

To harness these benefits of CDS-based QM, we have previously implemented a prototype CDS-QM framework in which a standards-based CDS Web service is leveraged across a population to enable QM.[6] This earlier prototype was able to successfully automate the evaluation of an inpatient quality measure (SCIP-VTE-2) with a recall of 100% and a precision of 96%.[6]

Need for optimizing execution performance. Despite the benefits of CDS-based QM, earlier research has identified execution performance as a critical challenge to operationally deploying such an approach at scale. For example, our earlier prototype of this approach[6] required approximately 5 seconds to process a measure group for a single patient. Similarly, the CDS Consortium has previously reported that a similar CDS Web service approach to patient evaluation required a similar amount of time to process.[7] While reasonable for QMs with limited population sizes, such as an inpatient quality measure which may only apply to several hundred patients at any given time, this approach is not scalable for QMs that apply to much larger patient populations, such as outpatient quality measures related to preventive care. For example, at 5 seconds per patient per measure group, processing 22 outpatient quality measure groups for 370,000 patients (e.g., to assess all patients seen within the past year at our health care system) would take over a year to process. Execution time is also an important consideration for real-time CDS as well as population-based CDS involving large numbers of patients.

At University of Utah Health Care (UUHC), our objective was to implement a CDS-QM framework for enterprise deployment, including for outpatient quality measures. To enable such enterprise use of this promising approach to QM, we needed to be able to evaluate a large number of QMs for very large patient populations in less than 24 hours (i.e. within one business day). This manuscript describes the performance optimizations that were implemented in order to meet this business need, such that the processing time was decreased approximately three orders of magnitude, from approximately 5 seconds per measure group per patient to 2 milliseconds per measure group per patient. By describing these performance optimizations and their relative impact, we seek to facilitate others' operational use of a CDS-QM approach for large-scale quality measurement and population health management.

Methods

Study Context

This study was conducted to meet operational quality measurement needs at University of Utah Health Care (UUHC), which serves as the Intermountain West's only academic healthcare system and includes four hospitals, 10 community clinics, over 10,000 employees, and over 1,200 physicians. Approximately 370,000 patients are seen at UUHC in any given year. This study was approved by the University of Utah IRB (protocol # 00080838).

The CDS-QM framework was initially applied at UUHC for calculating inpatient quality measures such as the Patient Safety Indicators (PSI) from the Agency for Healthcare Research & Quality (AHRQ) and Surgical Care Improvement Project (SCIP) measures from the Joint Commission. Because only several hundred hospitalized patients would be relevant for these measures at any given time, execution performance did not need to be optimized for evaluating these measures. The initial CDS-QM framework developed for inpatient quality measurement purposes represents the Phase I framework described below.

Subsequently, there was an institutional need to evaluate Healthcare Effectiveness Data and Information Set (HEDIS) outpatient quality measures from the National Committee for Quality Assurance. HEDIS measures are used by more than 90 percent of U.S. health insurers as a measure of quality and encompass measures such as the frequency of prenatal care visits compared to established standards and the degree of glycemic control in patients with diabetes mellitus.[8] HEDIS measures are calculated based on a combination of administrative/billing data and EHR data such as laboratory results, vital signs, immunization records, and procedure records.

At UUHC, the HEDIS measures had been calculated manually, in part due to the highly complex nature of many of the measures. For example, to calculate the number of prenatal visits that occurred in the first trimester of pregnancy, HEDIS defines 17 distinct patterns of eligibility. For this initiative, 22 measure groups (e.g., comprehensive diabetes care) encompassing 78 individual measures (e.g., % of members 18-75 years of age who had proper Hemoglobin A1c testing) were prioritized for automation.

The initial focus for the CDS-QM effort was to evaluate HEDIS measures for approximately 20,000 patients whom our commercial health plans considered to be primarily managed by UUHC clinicians. The platform developed to handle this scope of evaluation is reflected in the Phase II CDS-QM framework below.

Subsequently, the University of Utah Medical Group decided to base clinician compensation in part on their performance on HEDIS quality measures. This increased the applicable patient population to the approximately 370,000 patients seen at UUHC at least once within a one year timeframe. The platform developed to handle this scope of evaluation represents the Phase III CDS-QM framework below. In all cases, the business need was for the evaluation of all HEDIS quality measures to be completed for all relevant patients within one business day.

Required Computational Tasks

The CDS-QM framework was required to perform three primary computational tasks, each of which served as targets for performance optimization. The first task was data collection, in which patient data were retrieved from the data warehouse. In all phases, query parameters were set to restrict the data to those relevant for the required quality measure evaluations. The second task was data transformation, in which the collected data were converted into the Health Level 7 (HL7) Virtual Medical Record (vMR) data format, which is a standard data model for CDS.[9] Finally, the third task was data evaluation and storage, in which the OpenCDS decision support Web service[10] generates quality measurement assessments based on the vMR data, which are in turn stored in the data warehouse for analysis and reporting.

Qualitative Description of CDS-QM Framework by Phase

As described above, the business need for the CDS-QM framework evolved over three phases: a framework needed for evaluations of less than 1,000 patients at a time, evaluations of approximately 20,000 patients at a time, and evaluations of approximately 370,000 patients at a time. Accordingly, the framework was adapted in a phased approach. Each of these phases is described with respect to its approach to the three computational tasks.

Identification of Challenges and Solutions

The challenges encountered in optimizing execution performance are described, along with optimization solutions devised. Many different optimization approaches were implemented, but only those with significant impact are discussed here.

Scalability Analysis for Phase III Framework

For the final, phase III CDS-QM framework, we measured the time required for evaluating 22 HEDIS quality measure groups for increasingly large patient populations. The 22 HEDIS quality measure groups included 78 individual quality measurements. These analyses were conducted for 1 patient, 1,000 patients, 20,000 patients, and 370,000 patients.

Results

Qualitative Description of CDS-QM Framework by Phase

The overall CDS-QM process during phases I, II, and III are summarized in Figures 1, 2, and 3 respectively and described below. Moreover, the Table summarizes the optimization challenges, corresponding solutions, and impact for phases II and III. These challenges, solutions, and impact are also discussed in the framework descriptions below.

Phase I CDS-QM Framework

Overall Approach and Performance

In this phase, all three computational tasks were conducted on a patient-by-patient basis in sequence. The CDS-QM framework averaged approximately 5 seconds per patient per measure group, with almost all of the time required for data collection.

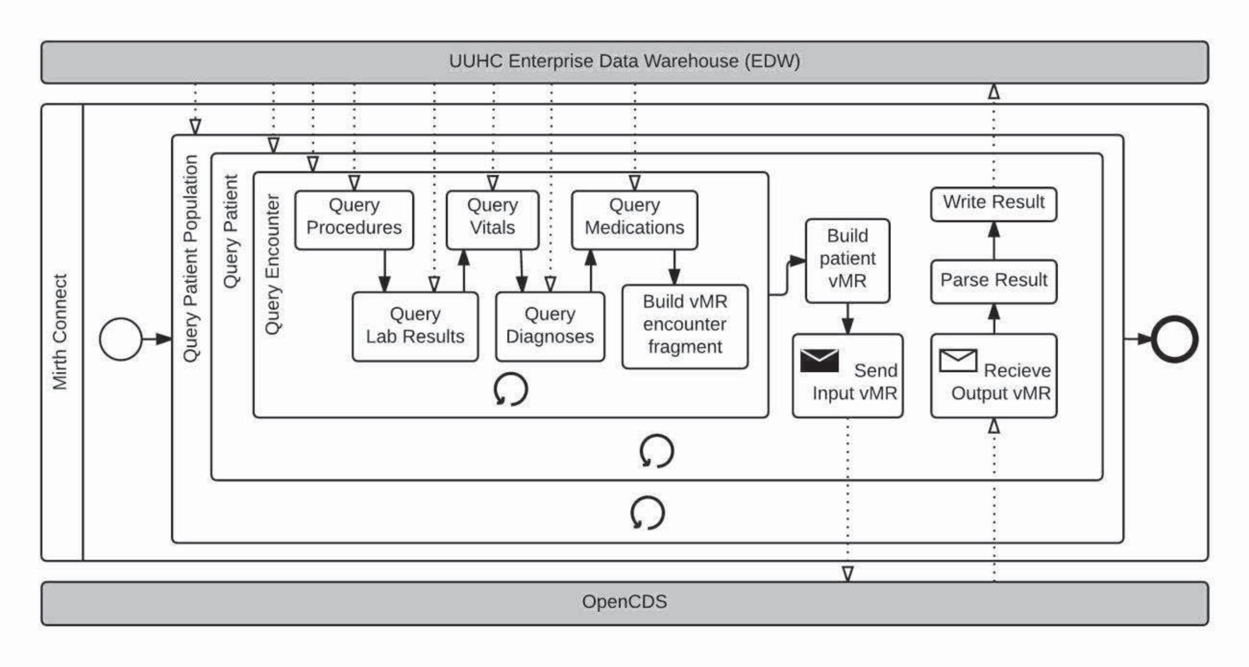

Figure 1. Phase I CDS-QM Process.

Phase I, Task 1: Patient Data Collection

In this phase, data were collected patient-by-patient, encounter-by-encounter using an integration engine called Mirth Connect. The integration engine directly queried the data warehouse and provided a scripting-based platform for accessing and transforming the patient data. An initial query identified the patient cohort to be analyzed. Then, for each identified patient, all subsequent tasks were conducted in sequence on a patient-by-patient basis. With regard to patient data collection, these patient-specific tasks included retrieval of demographic data and retrieval of relevant encounters. Then, for each relevant encounter, relevant associated data, such as for procedures, lab results, diagnoses, medications, and vital signs were retrieved in a sequential, repeating manner. This strategy produced a nested looping process where every nested query was run once for each patient and for each encounter. The collected data were held in volatile memory.

Phase I, Task 2: Data Transformation

After all queries were completed for an individual patient, the collected data were transformed into HL7 vMR objects. Data transformation occurred sequentially, one patient at a time.

Phase I, Task 3: Data Evaluation and Storage

The data were sent using an HL7 Decision Support Service-compliant interface[11], specifying which quality measures should be used to process the patient vMR. The results for each specified quality measure were returned to and parsed by the integration engine and written back to the data warehouse. This was also done sequentially, one patient after the other.

Phase II CDS-QM Framework

Challenges from Previous Phase, Overview of Solutions Devised, and Performance

There were two main challenges from the Phase I framework. First, data retrieval was a clear rate limiting step, such that the approach clearly could not scale to even tens of thousands of patients. A second challenge was that volatile memory limitations of the computing hardware prevented the simultaneous processing of a large number of patients.

To address these challenges, we altered the database queries to obtain data for the entire cohort of interest once, instead of for each patient sequentially. We also parallelized data queries and stored vMR fragments (e.g., vMR procedure fragments, vMR medication fragments) on disk to allow this parallelization within volatile memory constraints. This phase of the CDS-QM framework allowed approximately 20,000 patients to be evaluated for 22 HEDIS quality measure groups in approximately 8 hours, or an average of about 70 milliseconds per patient per measure group.

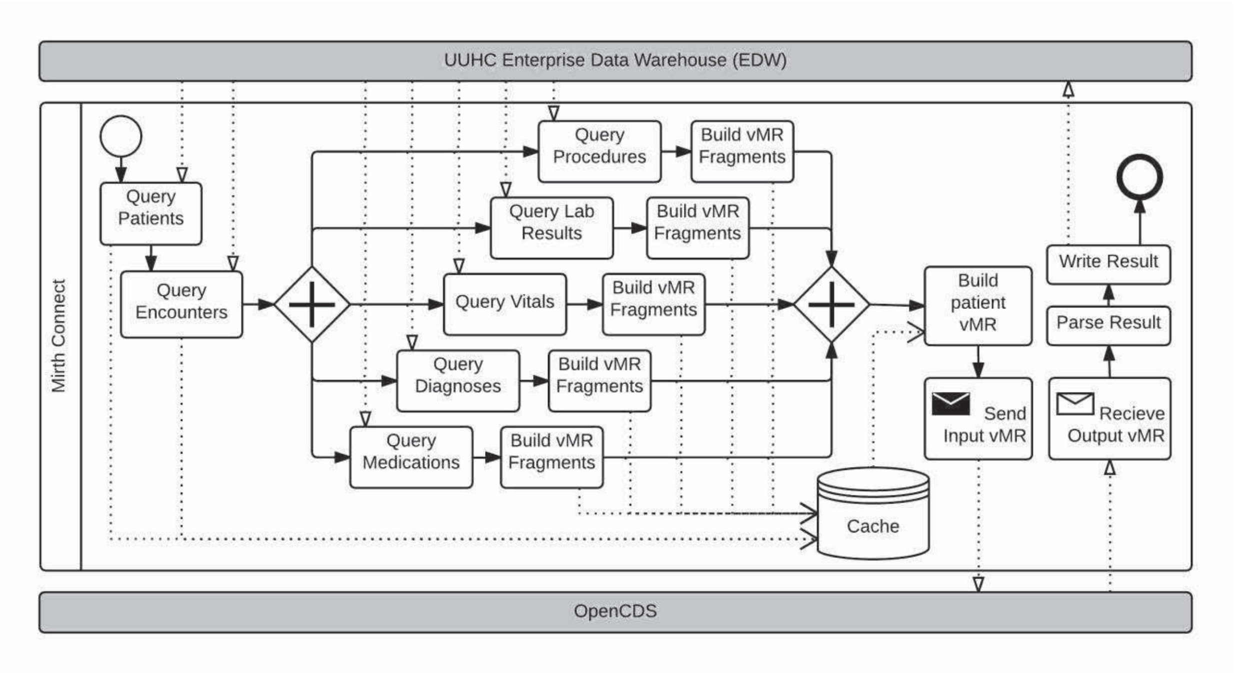

Figure 2. Phase II CDS-QM Process.

Phase II, Task 1: Patient Data Collection

The step for collecting patient data was first optimized by modifying and optimizing the query strategy to retrieve data for all patients in the measurement cohort rather than for a single patient per query. As long as source tables were properly indexed, most queries for the population data were nearly as fast as the same queries for a single patient. As a result, significant performance gains were realized after implementing this population-based query approach.

The result sets for these queries were large, so they could not be stored in volatile memory and hence were cached into a temporary schema. The cached results were later processed using a result set cursor, so that only a small portion of the result set was brought into volatile memory at any given time. We also found that our queries could run independently of each other, and hence modified our query strategy to include running the queries in parallel.

Phase II, Task 2: Data Transformation

The data transformation step was integrated into the data collection process. Fragments of the vMR were created for each relevant data element type (e.g., procedures, medications, lab results), and these fragments were stored in the temporary data cache as character large objects (CLOBs)[12]. The vMR fragments were then compiled with the demographic and encounter data from the temporary cache and transformed into individual patient vMRs.

Phase II, Task 3: Data Evaluation and Storage

Directly following vMR creation, each vMR was sent to the CDS Web service for processing. This step remained unmodified from the Phase 1 implementation.

Phase III CDS-QM Framework

Challenges from Previous Phase, Overview of Solutions Devised, and Performance

There were two main challenges from the Phase II framework. First, storing the transformed vMR fragments on disk became rate-limiting, due to the relatively slow speed of writing CLOBs into the data warehouse and reading from it. Also, for very large runs (e.g., 370,000 patients), this approach could not scale to the amount of storage space available in the data warehouse for this project. A second challenge was the sequential nature of vMR creation and evaluation. Whereas data retrieval was the clear bottleneck in Phase I, data storage (caching) and evaluation became the dominant bottlenecks in Phase II.

To address these challenges, we first stored the raw patient data into a standardized staging schema. For example, instead of storing a vMR XML fragment for procedures as a CLOB, the source data points required to generate this vMR XML fragment were stored in an indexed table with non-CLOB data types such as integers and dates. As a second key solution, we parallelized the vMR creation using multi-threaded processing techniques and used multiple servers, each with a deployed instance of OpenCDS, for evaluation.

Beyond these infrastructure-level optimizations, we also sought to optimize the execution performance of OpenCDS itself. We added the ability to process the set of requested measure groups in parallel. We also improved the performance of multiple requests by scaling the performance by the number of CPU cores (processors) available on the system. This current phase of the CDS-QM framework allows approximately 370,000 patients to be evaluated for 22 HEDIS quality measure groups in less than 5 hours, or an average of about 2 milliseconds per patient per measure group. Of note, continuous validation of CDS-QM results was performed against a sample of manual chart audit results. Systematic improvement based on this continuous validation process has resulted in a system with validity demonstrated to be equivalent to, and in some cases better than, manual QM analysis.

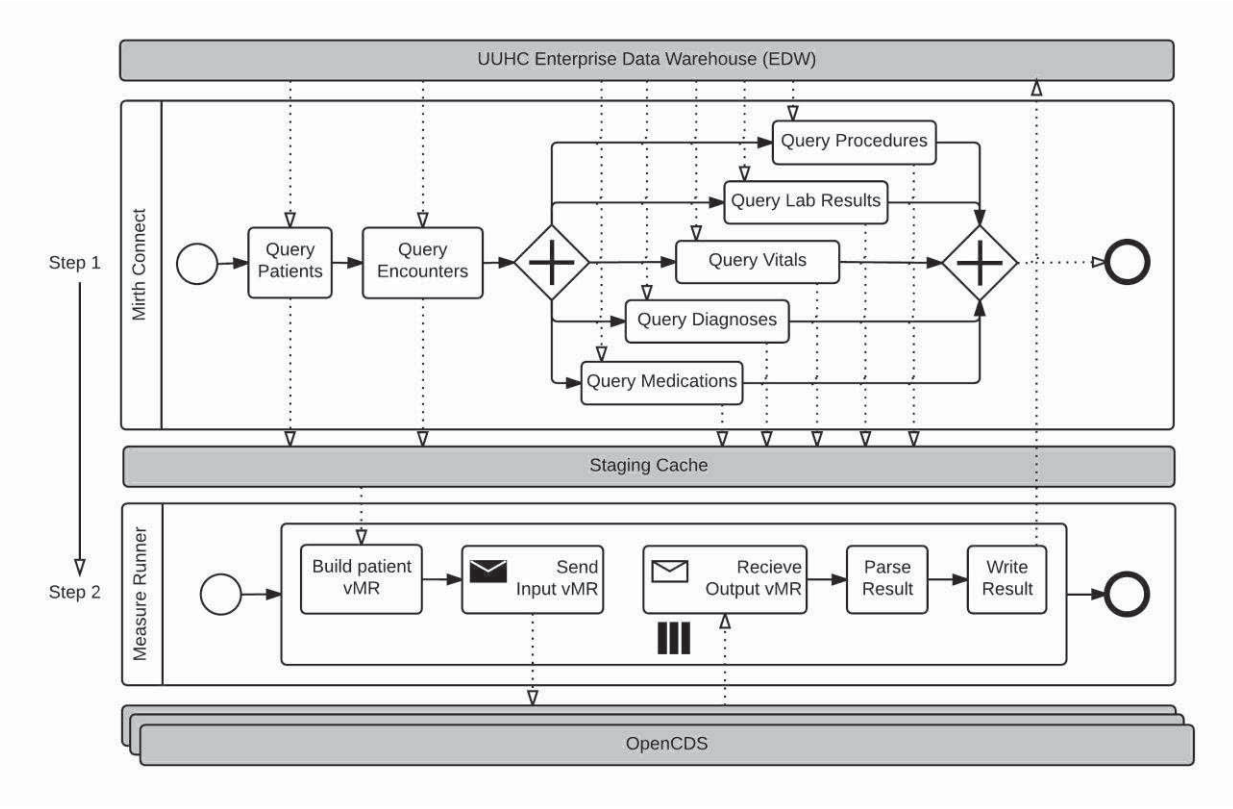

Figure 3. Phase III CDS-QM Process.

Phase III, Task 1: Patient Data Collection

Data collection continued to be conducted using population-based, parallel queries. However, instead of storing the collected data as pre-transformed vMR fragments in CLOBs, the source data elements were stored in their native data formats in a staging database schema.

Phase III, Task 2: Data Transformation

Staging the raw data, as opposed to the vMR fragments, facilitated the decoupling of the transformation task from the data collection task. Another subsystem, hereafter referred to as MeasureRunner, was developed to transform the data from the staging schema to individual patient vMRs. This process allowed the parallel creation of multiple patient vMRs, which in turn enabled the maximal use of available system resources.

Phase III, Task 3: Data Evaluation and Storage

MeasureRunner enabled the parallelization and batch evaluation of vMRs using the OpenCDS decision support service. Multiple instances of the decision support service were deployed to allow parallel processing. Responses were parsed and stored asynchronously as they were returned by the decision support service instances.

Challenges and Solutions

Table 1 summarizes the challenges encountered during this multi-phase optimization process, as well as the key solutions that were found to address these challenges.

Table 1. Key Challenges and Solutions

Phase	Primary Challenges Addressed from Previous Phase	Key Solutions
Phase II	• Data retrieval was clear bottleneck and did not scale to large populations • Limitations in the amount of available volatile memory prevented simultaneous processing of large populations	• Altered queries to return data for entire cohort instead of per patient • Parallelized data queries • vMR fragments stored on disk to allow parallelization
Phase III	• Storage of vMR fragments on disk would not scale because of limited disk space and relatively slow speed of reading and writing CLOBs • vMR creation and evaluation was sequential	• Stored raw patient data rather than vMR fragments into a standardized staging schema to improve caching and transformation performance • Parallelized vMR creation and evaluation using multi-threading and multiple servers

CLOB = Character Large Object. vMR = Virtual Medical Record.

Scalability Analysis

Figure 4 shows the execution time for the Phase III CDS-QM framework when evaluating increasingly large cohort sizes for 22 HEDIS quality measure groups. As noted, the process scales up well for large cohort sizes, with average completion times per patient per measure group actually decreasing with large cohort sizes.

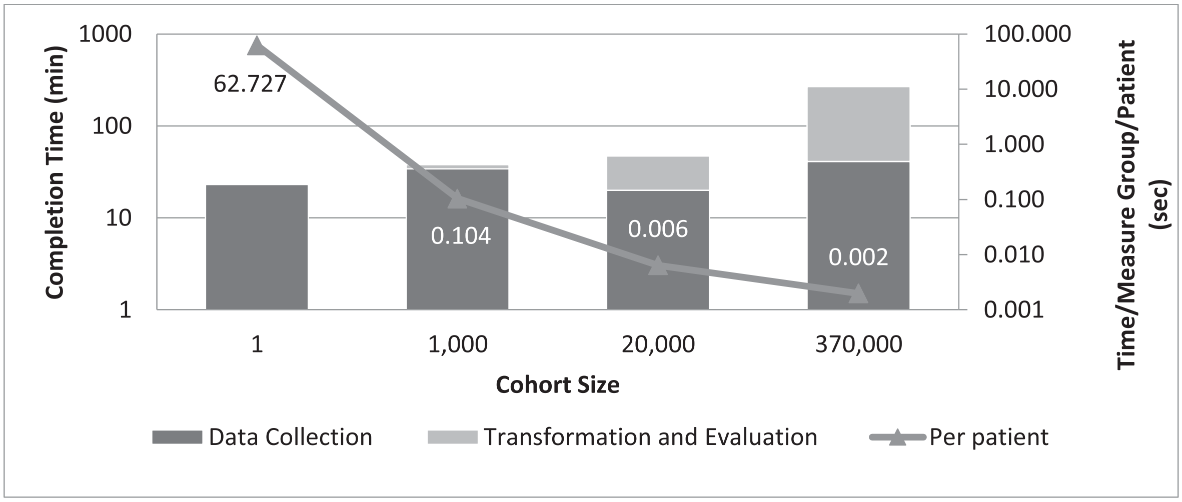

Figure 4. Phase III framework performance times with cohort size and per measure per patient evaluation time

Discussion

Summary of Findings

A standards-based CDS-QM framework was enhanced across two phases in this study to enable the evaluation of increasingly large patient cohorts in a timely manner. Through key strategies including population-based data collection, parallel processing, and the use of intermediate staging tables, the CDS-QM framework achieved a performance improvement of approximately three orders of magnitude. The current CDS-QM framework is able to process 22 quality measure groups for all patients seen at least once in the past year in an academic health system in less than 5 hours.

Impact

In terms of local impact, the CDS-QM framework now allows complex clinical quality measures to be evaluated for all applicable patients within one business day, as opposed to depending on estimations based on chart audits that required weeks or months to complete. Consequently, the CDS-QM framework is enabling a number of clinical quality improvement initiatives, including those with explicit ties to physician compensation.

In terms of impact beyond our institution, we speculate that the lessons learned from this study will facilitate efforts by others to deliver operational CDS and QM using a common CDS-QM framework, thereby achieving the benefits of semantic consistency and implementation efficiency. In particular, we hope that the open-source nature of the relevant software components will encourage the adoption and community improvement of the CDS-QM framework.

Limitations and Strengths of Study

One significant limitation of this study is that execution timings were not systematically recorded, as the development of each phase of the CDS-QM framework was conducted for operational rather than research purposes. However, we believe that the exact magnitude of benefit gained from each strategy is less important than the overall magnitude of benefit gained from the combination of all strategies. A second limitation of this study is that the execution performance of the CDS-QM framework has not yet been evaluated at other institutions. Finally, the data retrieval component of our implementation is currently institution-specific. While some customization is likely inevitable for institution-specific data retrieval (e.g., to account for differing database schemas), our goal is to generalize this data retrieval component so that institution-specific customizations can be implemented with high efficiency.

An important strength of this study is that the framework described has been shown to be sufficiently robust for operational, enterprise-level use. Indeed, the analysis described in the Phase III framework is now routinely being conducted for operational clinical use. A second strength of this study is that it utilizes open-source, standards-based software components that are freely available. A third strength is the scalability analysis, which indicates that the approach scales well to increasing cohort sizes. A final strength of this approach is that it is a general-purpose framework that can be adopted to other quality measures as well as to other large-scale analysis needs, including population health management.

Future Directions

The core CDS Web service is already freely available through www.opencds.org. We intend to make the other components of the CDS-QM framework also freely available after re-factoring the components for general use. We also intend to continually enhance the CDS-QM framework and its knowledge content based on the operational needs of our healthcare system and of other stakeholders making use of this framework.

Conclusion

Leveraging a standards-based CDS infrastructure for quality measurement has important benefits, including semantic consistency and implementation efficiency. We hope that the performance optimization methods identified in this study will enable such a CDS-QM approach to be more widely leveraged in operational settings to improve health and health care.

Acknowledgements

This study was funded by University of Utah Health Care. KK is currently or recently served as a consultant on CDS to the Office of the National Coordinator for Health IT, ARUP Laboratories, McKesson InterQual, ESAC, Inc., JBS International, Inc., Inflexxion, Inc., Intelligent Automation, Inc., Partners HealthCare, the RAND Corporation, and Mayo Clinic. KK receives royalties for a Duke University-owned CDS technology for infectious disease management known as CustomID that he helped develop. KK was formerly a consultant for Religent, Inc. and a co-owner and consultant for Clinica Software, Inc., both of which provide commercial CDS services, including via the use of a CDS technology known as SEBASTIAN that KK developed. KK no longer has a financial relationship with either Religent or Clinica Software. KK has no competing interests related to OpenCDS, which is freely available to the community as an open-source resource. All other authors declare no conflict of interest.

References

1. Goldstein MK, Tu SW, Martins S, et al. Automating performance measures and clinical practice guidelines: differences and complementarities. AMIA Annu Symp Proc. 2014;37–8.

2. Haggstrom DA, Saleem JJ, Militello LG, Arbuckle N, Flanagan M, Doebbeling BN. Examining the relationship between clinical decision support and performance measurement. AMIA Annu Symp Proc. 2009 Jan; 2009:223–7.

3. Forrest CB, Fiks AG, Bailey LC, et al. Improving adherence to otitis media guidelines with clinical decision support and physician feedback. Pediatrics. 2013 Apr; 131(4):e1071–81.

4. Standards & Interoperability Framework. Clinical Quality Framework Initiative. Available from: http://wiki.siframework.org/Clinical+Quality+Framework+Initiative

5. Eisenberg F, Lasome C, Advani A, Martins R, Craig PA, Sprenger S. A study of the impact of Meaningful Use clinical quality measures. American Hospital Association. 2013. Available from: http://www.aha.org/research/policy/ecqm.shtml

6. Kukhareva P, Kawamoto K, Shields DE, et al. Clinical Decision Support-based Quality Measurement (CDS-QM) framework: prototype implementation, evaluation, and future directions. AMIA Annu Symp Proc. 2014;825–34.

7. Paterno MD, Goldberg HS, Simonaitis L, et al. Using a service oriented architecture approach to clinical decision support: performance results from two CDS Consortium demonstrations. AMIA Annu Symp Proc. 2012 Jan; 2012:690–8.

8. National Committee for Quality Assurance. HEDIS & Performance Measurement. Available from: http://www.ncqa.org/HEDISQualityMeasurement.aspx

9. Health Level 7. Virtual Medical Record (vMR). Available from: http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_%28vMR%29

10. OpenCDS Home Page. Available from: http://www.opencds.org/

11. Health Level 7. Decision Support Service (DSS). Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12

12. Oracle. CLOB data type. Available from: http://docs.oracle.com/javadb/10.10.1.2/ref/rrefclob.html